REMARKS

Claims 1-23 are currently pending in the subject application and are presently under consideration. Claims 1, 7-14, 16, 17, 22, and 23 have been amended as shown on pages 2-6 of the Reply.

Applicant's representative thanks the Examiner for the courtesies extended during the telephonic interview conducted on October 16, 2007. During the interview, the Examiner made several helpful suggestions for amendments intended to improve the consistency of terminology used throughout the claim set. These suggestions have been incorporated into the present claim set found herein. The Examiner also indicated that the amendments to independent claims 1 and 14 would most likely overcome the rejections under 35 U.S.C. §101(a). Participants did not discuss the interpretations of the cited art reference, since the participating Examiner had not personally prepared the present Office Action and was not familiar with the cited references. However, the Examiner indicated that she would consider the arguments fully upon receiving the submitted Reply.

Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I. Objection of Claim 1

Claim 1 is objected to because the Examiner contends the claim is grammatically incorrect. The objected terminology has been amended in accordance with the Examiner's preferences.

II. Rejection of Claims 9-12 Under 35 U.S.C §112

Claims 9-12 stand rejected under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claim 9 has been herein amended to address the Examiner's concerns regarding the phase "a workplace."

The Examiner also takes issue with the fact that "a unique identifier" is recited in both claims 10 and 11, and contends that it is unclear if these unique identifiers are intended to be the same or different. Both claims depend from claim 9, and each respectively describes the state information that can be stored in the two scenarios presented in claim 9; namely, *all files and* *folders* (claim 10), or *individual files* (claim 11). Hence, claim 10 recites *a unique identifier for* ***all files and folders***, while amended claim 11 recites *a unique identifier for* ***an individual file***. These clearly denote two different identifiers by virtue of their respective associated entities. Applicant's representative points out that, in general, multiple entities may each have "a unique identifier," and referring to each one as a "unique identifier" should not cause concern, since the identifiers are distinguished by their associated entities. Nevertheless, claims 10 and 11 have been amended to more clearly establish that these two claims respectively refer to the two scenarios disclosed in claim 9; namely *all files and folders* and *individual files*.

In view of the aforementioned amendments, it is respectfully requested that this rejection be withdrawn.

III. Rejection of Claims 1-12, 14 and 15 Under 35 U.S.C. §101

Claims 1-12, 14 and 15 stand rejected under 35 U.S.C. §101 because the Examiner contends the claimed invention is directed to non-statutory subject matter. Independent claims 1 and 14 have been amended to address the Examiner's concerns in connection with this rejection. Accordingly, it is respectfully requested that this rejection be withdrawn.

IV. Rejection of Claims 1-6, 9-13, 16-19 and 21 Under 35 U.S.C. §102(b)

Claims 1-6, 9-13, 16-19 and 21 stand rejected under 35 U.S.C. §102(b) as being anticipated by Parrish (5,752,245). It is respectfully submitted that this rejection should be withdrawn for at least the following reasons. Parrish fails to disclose each and every feature of applicant's claimed subject matter.

> For a prior art reference to anticipate, 35 U.S.C. §102 requires that "each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950 (Fed. Cir. 1999) (*quoting Verdegaal Bros., Inc. v. Union Oil Co.*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987)).

The subject claims relates to a system and method for archiving software development works in progress within a version control system. Developers working on a software project in a local workspace can submit intermediate versions of the software to a storage repository using a concept known as shelving. When a developer decides to shelve a software work in progress, he or she has can opt to have the local versions of the associated software files restored to their unaltered versions as part of the shelving process. In other words, shelving a software work can archive any pending changes made since the last restored version of the software, and can also remove those changes from the local version of the software. In particular, amended independent claim 1 recites, *a shelving component that captures a current state of an intermediate software design containing pending changes from a private workspace,* ***the shelving component removes the pending changes from the software design on the private workspace after the current state has been captured.***

Parrish does not disclose these features of the subject claims. The cited reference relates to a distributed database system used to maintain version information for program components. Program developers establish a workspace on a client terminal and retrieve drafts of all components of a program configuration from the distributed database servers. The workspace then assembles the complete source code so that development can continue on the software code. The developer can subsequently send drafts of the modified software components to the distributed server system for storage. However, Parrish nowhere teaches that the process of archiving a version of a software project *also removes pending changes from the local workspace version of the software* as disclosed by the subject claims. This feature represents an advantage over the cited reference, in that it offers developers the convenience of easily shelving an interrupted development task and undertaking a different task on the same software project without mixing the two development threads.

The subject claims also teach that shelved software versions can be unshelved to a local workspace. Unshelving an archived work restores the local workspace to a previously archived state so that development can continue. Permissives associated with the unshelve action are a function of who owns the archive being unshelved. For example, when an archived work is unshelved by the owner of the work, the archive can be deleted from the server as part of the unshelve action. Alternatively, when a non-owner unshelves another user's work, the archive is preserved on the server. In particular, amended independent claim 16 (and similarly amended claim 23) recites, *unshelving the version of software to a local workspace in accordance with the state, the version of software is* ***deleted from the version control system when unshelving is initiated by an owner of the software****, and the version of software is* ***preserved on the version***

control system when unshelving is initiated by a non-owner of the software. In contrast, while Parrish teaches that archived software drafts can be retrieved to a local workspace, the act of retrieval preserves the active archived files on their associated servers *in all cases*, without consideration of permissives (see especially column 19 lines 8-18, which explicitly discloses that, although components that are no longer used are deleted after a retrieval, normal components are left unchanged). By removing the archived software from the server when the owner unshelves the archive, the subject claims can prevent the accumulation of unwanted software drafts on a server, an advantage not provided by the cited reference. Additionally, granting non-owners the ability to unshelve software belonging to other users while maintaining the archived copy can facilitate flexible work sharing.

In addition, the subject claims teach that unshelving software files to a local workspace that already contains pending changes can facilitate merging the pending changes with the unshelved version of the files. In particular, amended claim 13 (and similarly amended claim 17) recites, *the unshelving component merges unshelved changes with changes pending on the private workspace when the unshelve operation is initiated.* Parrish does not contemplate merging archived changes with pending changes on a local workspace. Rather, the system disclosed by Parrish requires a user to begin with an *empty local workspace* before retrieving program components from their associated servers (see column 17 lines 13-22). Hence, the cited reference does not offer the ability to merge a retrieved software draft with changes that are pre-pending on a local workspace, since the reference teaches that there can be no local changes pending at the time a retrieval is initiated.

In view of at least the foregoing, it is respectfully submitted that Parrish fails to disclose each and every element set forth amended independent claims 1 and 16 (and all claims depending there from) and therefore fails to anticipate applicant's claimed subject matter. It is therefore requested that this rejection be withdrawn.

V. Rejection of Claims 14, 15, and 22 Under 35 U.S.C. §102(b)

Claims 14, 15, and 22 stand rejected under 35 U.S.C. §102(b) as being anticipated by Ziebell (6,385,768). However, Ziebell fails to disclose all features of the subject claims. In particular, and with respect to the aforementioned feature of removing pending changes from a local workspace version of the software when the software is archived, amended independent claim 14 (and similarly amended independent claim 22) recites, *means for developing pending changes on non-finalized software on a local workspace; means for archiving the non-finalized software with pending changes to a version control system; means for capturing one or more states associated with the non-finalized software; [and]* ***means for removing the pending changes from the non-finalized software on the local workspace.***

Like Parrish, Ziebell is silent regarding these features. Ziebell relates to a version control program whereby various revisions of a software project are stored in an architecture of branches and streams implemented on a server. Revisions to a software project can be checked into the archive system as a series of "deltas," which contain information about the updates made to a source revision. These deltas can be selectively applied to a software revision stored on a stream of a revision tree to create a new revision on the stream. However, Ziebell does not teach that pending changes on a local copy of non-finalized software are removed when the software is archived. Although Ziebell discloses that such updates or changes to software version can be stored as deltas on an archive system, the cited reference nowhere teaches that such changes are *removed from the local version of the software.*

In view of at least the foregoing, it is respectfully submitted that Ziebel fails to teach or suggest each and every feature set forth amended independent claims 14 and 22 (and claim 15, which depends from claim 14) and therefore fails to anticipate applicant's claimed subject matter. It is therefore requested that this rejection be withdrawn.

VI. Rejection of Claims 7, 8, and 23 Under 35 U.S.C. §103(a)

Claims 7, 8, and 23 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Parrish (5,752,245), in view of Ziebell (6,385,768). It is respectfully submitted that this rejection should be withdrawn for at least the following reasons. Neither Parrish nor Ziebell, individually or in combination, disclose all aspects set forth in the subject claims.

Amended claim 8 recites, *the unshelving component allows* ***removing the state that is stored on a server*** *or preserving changes shelved on the server in order that the changes are available for unshelving by other users.* As conceded in the Office Action, Parrish does not disclose these features of the subject claims. However, contrary to the Examiner's contentions, Ziebell fails to remedy this deficiency of Parrish. In addition to the features of Ziebell discussed *supra*, the cited reference also teaches that revisions can be checked out and modified prior to having the modifications checked into the system as deltas. However, Ziebell does not teach that the revisions can be *removed from the server* when checkout is performed. Indeed, allowing a user to remove a revision from the branching architecture taught by Ziebell could conceivably result in stability issues, since dependency relationships in the form of branches and streams exist between the various revisions. The portion of Ziebell cited by the Examiner merely discloses that a revision is added to a logical "change" unit when the revision is checked out, to facilitate tracking the components that have been modified. This in no way suggests *removing* the revision from the server. In contrast, the subject invention need not rely on a branching architecture, and in fact seeks to provide a simplified alternative to the branching techniques exemplified by Ziebell, which can lead to relatively complicated file structures in the version repository. The subject invention's lack of a complicated branching structure affords a cleaner file system in which to archive software revisions, assisted by the aforementioned feature of removing software states from the server when the software is unshelved. Ziebell, by teaching a *branching* method for archiving and tracking software versions, in fact teaches away from applicant's claimed invention.

Moreover, claim 7 depends from amended independent claim 1, and amended claim 23 depends from amended independent claim 22. As discussed *supra* with regard to the independent claims, neither cited art reference discloses *removing pending changes from non-finalized software on a local workspace when the software is shelved.*

In view of at least the foregoing, it is respectfully submitted that Parrish and Ziebell, individually or in combination, do not teach or suggest all features of applicant's claimed invention as recited in independent claims 1 and 22 (and claims 7, 8, and 23, which depend there from). It is further submitted that the branching methods of Ziebell teach away from the methods disclosed in the subject claims. It is therefore requested that this rejection be withdrawn.

VII. Rejection of Claim 20 Under 35 U.S.C. §103(a)

Claim 20 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Parrish (5,752,245). However, claim 20 depends from amended independent claim 16. As discussed *supra* with respect to that independent claim, both Parrish and Ziebell fail to disclose *deleting software from the version control system when unshelving is initiated by an owner of the* *software*. It is therefore respectfully requested that this rejection be withdrawn with respect to claim 20.

CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [MSFTP572US].

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,
AMIN, TUROCY & CALVIN, LLP

/Himanshu S. Amin/
Himanshu S. Amin
Reg. No. 40,894

AMIN, TUROCY & CALVIN, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731